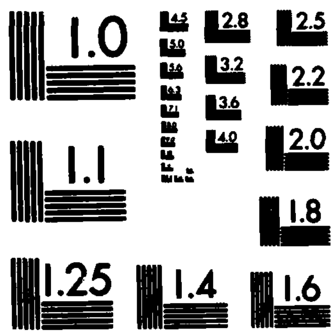
AD-A133 545   A SURVEY OF NETWORK RELIABILITY(U) CALIFORNIA UNIV BERKELEY OPERATIONS RESEARCH CENTER A AGRAWAL ET AL. JUL 83 ORC-83-5 AFOSR=81-0122   1/1

UNCLASSIFIED   F/G 9/2   NL

END



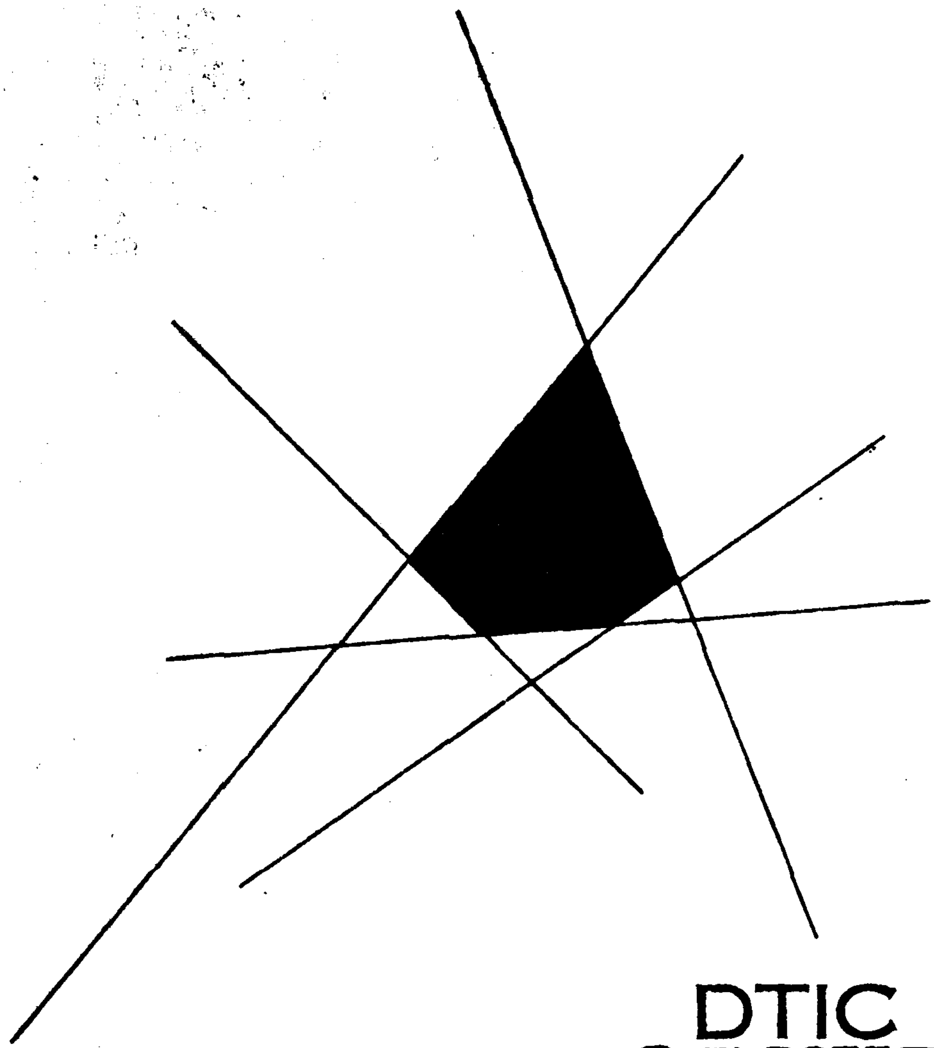
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

ORC 83-5
JULY 1983

A SURVEY OF NETWORK RELIABILITY

by
AVINASH AGRAWAL
and
RICHARD E. BARLOW



OPERATIONS RESEARCH CENTER

UNIVERSITY OF CALIFORNIA · BERKELEY

A SURVEY OF NETWORK RELIABILITY

by

Avinash Agrawal
Tymnet, Inc.
San Jose, California

and

Richard E. Barlow
Department of Industrial Engineering
and Operations Research
University of California, Berkeley

JULY 1983 ORC 83-5

This research was supported by the Air Force Office of Scientific Research (AFSC), USAF, under Grant AFOSR-81-0122 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ORC 83-5	2. GOVT ACCESSION NO. AD-A133 545	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A SURVEY OF NETWORK RELIABILITY		5. TYPE OF REPORT & PERIOD COVERED Research Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Avinash Agrawal		8. CONTRACT OR GRANT NUMBER(s) AFOSR-81-0122
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operations Research Center University of California Berkeley, California 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2304/A5 61102F
11. CONTROLLING OFFICE NAME AND ADDRESS United States Air Force Air Force Office of Scientific Research Bolling Air Force Base, D.C. 20332		12. REPORT DATE July 1983
		13. NUMBER OF PAGES 28
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

System Reliability
Graph Domination
Computational Complexity
Coherent Structures
Networks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(SEE ABSTRACT)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

We present a brief survey of the current state of the art in network reliability. We survey only exact methods; Monte Carlo methods are not surveyed.

Most network reliability problems are, in the worst case, NP-hard and are, in a sense, more difficult than many standard combinatorial optimization problems.

Although the above sounds very discouraging, there are in fact linear and polynomial time algorithms for network reliability problems of special structure.

We review general methods for network reliability computation and discuss the central role played by domination theory in network reliability computational complexity. We also point out the connection with the more general problem of computing the reliability of coherent structures. ~~[c.f. Barlow and Proschan (1981)].~~ The class of coherent structures contains both directed and undirected networks as well as logic (or fault) trees *without* not gates. This is a rich area for further research.

A SURVEY OF NETWORK RELIABILITY

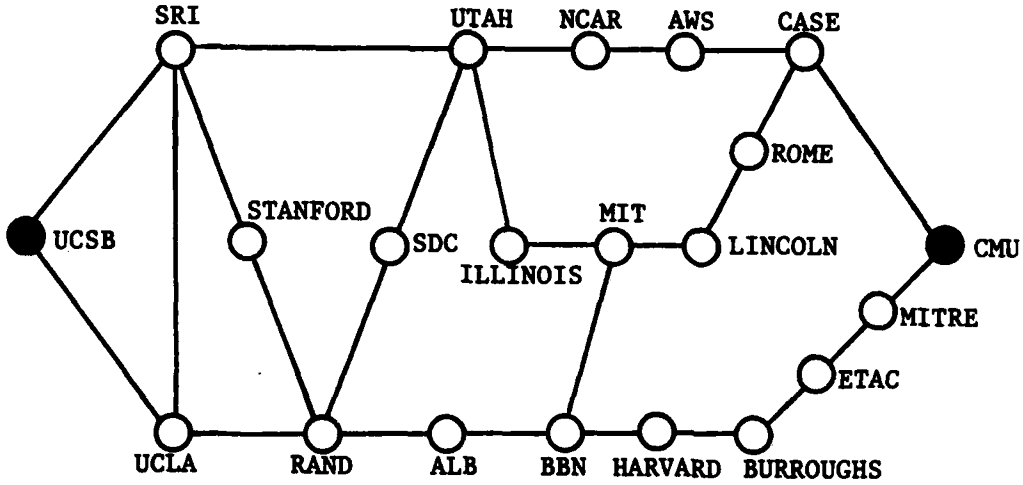
by

Avinash Agrawal and Richard E. Barlow

1. INTRODUCTION

Figure 1 is a well-known example of a computer communication network. A two-terminal reliability problem would be that of computing the probability that the distinguished node labelled UCSB can communicate with the distinguished node labelled CMU via some set of arcs or edges. Edges may be subject to failure. In this paper, the edge success probabilities are assumed known and the associated success events are assumed independent given those probabilities. A typical network reliability problem is to calculate *efficiently* the probability that a specified set of nodes can communicate with each other at a given time. Edges may be either directed or undirected, the direction, when it applies, will be denoted by an arrow on the edge.

In this paper, we present a brief survey of the current state of the art in network reliability. We survey only exact methods; Monte Carlo methods are not surveyed. There are many papers in this field, some give methods to simplify or solve the problems while others calculate the complexity of network reliability problems. In 1975, Rosenthal showed that certain fault tree and network reliability problems are inherently difficult and almost certainly have no fast algorithm. Since then, this class has grown to contain a number of other network reliability problems and some of the questions that he raised regarding other related problems have been answered.



ARPA COMPUTER NETWORK

FIGURE 1

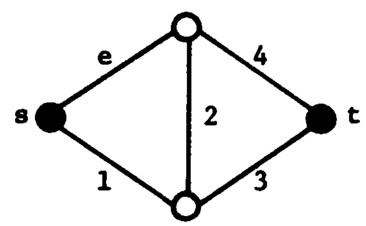
We begin with a very brief introduction to the theory of computational complexity and NP-completeness. More details can be found in Garey and Johnson (1979). An algorithm is a *polynomial time algorithm* if for a problem of size n, its running time is bounded by a polynomial in n. Any algorithm that is not a polynomial time algorithm is generally referred to as an *exponential time algorithm.* In combinatorics, the so-called "satisfiability" problem is in the class NP-complete, i.e., given an arbitrary Boolean expression in product of sums form, determine whether or not there exists an assignment of values TRUE or FALSE to the variables which makes the entire expression TRUE (a Boolean expression can be obtained for every network in terms of Boolean indicators for the edges and nodes). A problem P is said to belong to the class *NP-complete* if (i) given a purported solution its validity can be checked in polynomial time, (ii) the existence of an algorithm to solve P in polynomial time implies the existence of algorithms to solve the satisfiability problem in polynomial time. It is generally believed that no polynomial time algorithm exists for any of the NP-complete problems. Any problem not NP-complete but which can be proved to be at least as hard as NP-complete problems is known as an *NP-hard problem.*

Most network reliability problems are, in the worst case, NP-hard [Ball (1977 and 1980)]. Network reliability problems are, in a sense, more difficult than many standard combinatorial optimization problems. That is, given a tentative solution to a combinatorial problem, often its correctness can be determined in polynomial time. However, given a purported solution to a reliability problem, it cannot even be checked without computing the reliability of the network from the beginning.

Although the above remarks sound very discouraging, there are in fact linear and polynomial time algorithms for network reliability problems of special structure. These will be reviewed in Section 3. In Section 2, we review general methods for network reliability computation and discuss the central role played by domination theory in network reliability computational complexity. We also point out the connection with the more general problem of computing the reliability of coherent structures [cf. Barlow and Proschan (1981)]. The class of coherent structures contains both directed and undirected networks as well as logic (or fault) trees *without* not gates [cf. R. R. Willie (1978)]. This is a rich area for further research.

2. COMPUTATIONAL COMPLEXITY OF FACTORING ALGORITHMS: DOMINATION THEORY

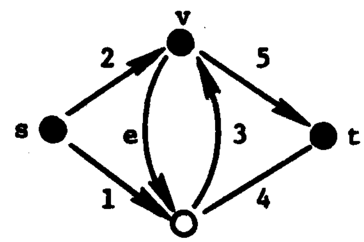
Consider a graph $G = (V,E)$ with vertex set V and edge set $E = \{1,2, \ldots, n\}$. Vertices do not fail, but at an instant of interest, an edge i has reliability p_i, independent of the states of other edges. Figure 2 is a network graph with edge set $E = \{1,2,3,4,e\}$. Since no arrows appear on edges, this is an undirected network. The graph, together with the distinguished nodes $K = \{s,t\}$, define a network reliability problem - namely, calculate the probability that s and t can communicate. The distinguished nodes also define the topology of the problem - namely, the family of minimal path sets, $\mathcal{P} = [\{1,3\},\{1,2,4\},\{e,4\},\{e,2,3\}]$. (A minimal path set is a minimal set of elements whose functioning implies that the distinguished nodes can communicate.) A different set of distinguished nodes would define a different topology.



UNDIRECTED TWO-TERMINAL NETWORK

FIGURE 2

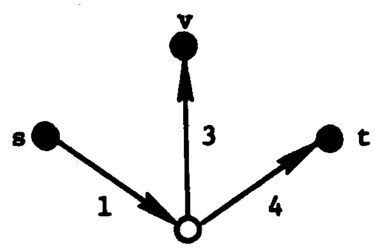
Figure 3 is a *rooted directed graph* with distinguished node set $K = \{s,t,v\}$. The "root" is vertex s, the unique node with no entering edges. The problem is to compute the probability that s can communicate with both v and t. The topology for this problem is defined by the family of minimal path sets $\mathcal{P} = [\{2,5\},\{1,3,5\},\{2,e,4\},\{1,3,4\},\{1,2,4\}]$.



DIRECTED NETWORK

FIGURE 3

Graphically, a minimal path for this problem is a "rooted tree." For example, the following acyclic rooted directed graph is a "tree" corresponding to Figure 3.



ROOTED TREE

FIGURE 4

Since, historically, not much emphasis has been placed in the reliability literature on computational complexity and because these problems are, in general, NP-hard, many different algorithms have been suggested to solve these problems. Many algorithms are based on minimal path sets and/or cut sets (a minimal cut set is a minimal set of elements whose failure implies that some distinguished nodes cannot communicate). In general, however, it is neither necessary nor desirable to find the family of minimal paths or cut sets in order to calculate network reliability.

The reliability of a graph, G, with distinguished nodes K, is the probability, $R_K(G)$, that all elements of at least one minimal path set are working or one minus the probability that all elements of at least one minimal cut set have failed. Note that $R_K(G)$ depends on the distinguished node set $K \subseteq V$ as well as G. Different methods exist to evaluate this probability. These methods are quite general and can be used relative to any system reliability problem. We shall roughly classify them into three main classes. Let A_i denote the event that all elements in the i^{th} minimal path set are functional and $\bar{A}_i$ denote the complement of this event. Let p be the number of minimal path sets.

(i) <u>The Inclusion-Exclusion Method</u>

$$R_K(G) = P\left[\bigcup_{i=1}^{p} A_i\right] = \sum_{i=1}^{p} P(A_i) - \sum_{i=1}^{n} \sum_{j<i} P[A_iA_j] + \ldots + (-1)^{p-1}P[A_1A_2 \ldots A_p] \; . \tag{2.1}$$

If there are p path sets, then this calculation involves $2^p - 1$ terms. In some cases, two different intersections of A_i's will correspond to the same event so that these different intersections of A_i's will have the same probability. If one intersection consists of an odd number of A_i's and another intersection consists of an even number of A_i's, their probabilities in the inclusion-exclusion expression will cancel. Satyanarayana and Prabhakar (1978) and Satyanarayana (1982) give algorithms that generate only the non-cancelling terms. In the *reduced* inclusion-exclusion expression, there will be a coefficient for the term corresponding to the event $A_1A_2 \ldots A_p$. (This event will in general correspond to several different intersections.) The coefficient for this term is called the

signed domination, $d_K(G)$. This coefficient is the number of odd *formations* of G minus the number of even formations of G . (A *formation* is a set of minimal path sets whose union constitutes the edge set E . It is odd (even) if it has an odd (even) number of minimal path sets in it.) The absolute value of this coefficient is called the *domination*, $D_K(G) = |d_K(G)|$. As we shall see, this number is a measure of the computational complexity of certain factoring algorithms for undirected networks. The network corresponding to Figure 2 has 6 formations of which 4 are odd and 2 are even so that $d_K(G) = 2$ in this case.

(ii) Sum of Disjoint Products

$$R_K(G) = P(A_1) + P(\bar{A}_1 A_2) + \dots + P(\bar{A}_1 \bar{A}_2 \dots \bar{A}_{p-1} A_p) \ . \qquad (2.2)$$

For p path sets, there are p terms but the time needed to generate each term may be exponential in p . Most methods based on Boolean techniques belong to this class. Such methods have been proposed by Abraham (1979), Fratta and Montanari (1973), Aggarwal, Misra and Gupta (1975), etc. One proposed by Lee (1979) for flow networks can easily be modified by using concepts of Satyanarayana (1982) to give the reliability of communication networks. This method, based on the idea of using backtracking and depth first search introduced by Gabow and Myers (1978), is the most efficient method in this class for the undirected all-terminal and the directed source-to-all-terminal problems; i.e., the distinguished set of vertices is the set of all vertices of the graph. The amount of computational work required is proportional to the number of spanning trees or rooted spanning arboresences. [See Deo (1974) for a detailed discussion of graph theory concepts.]

(iii) Pivotal Decomposition or Factoring

If $R_K(G \mid e)$ is the reliability of G under the condition that edge e is working and $R_K(G \mid \bar{e})$ the reliability of G under the condition that edge e is not working, then using the pivotal decomposition [cf. Barlow and Proschan (1981)],

$$R_K(G) = p_e R_K(G \mid e) + (1 - p_e) R_K(G \mid \bar{e}) . \tag{2.3}$$

$R_K(G)$ for any graph G can be computed by repeated application of this decomposition. Undirected graphs have some special properties that can be used to simplify this method. If the vertices are assumed to be working, then $R_K(G \mid e)$ is the same as $R_K(G_e)$ where G_e is the graph obtained from G by deleting edge e and merging its end points. Similarly, $R_K(G \mid \bar{e})$ is the same as $R_K(G - e)$ where $G - e$ is the graph with e deleted but no vertex deleted. These schemes have been discussed by a number of writers, among them, Moskowitz (1958) and Misra (1970). In the literature, the above relationship has been referred to as the *Factoring Theorem*. It is important to note that this method can be employed using the graph representation but *without* knowing the minimal path sets. However, unless some kind of probability reductions are performed (e.g., parallel and series reductions) after each pivot, this method will be equivalent to state space enumeration.

The Factoring Algorithm for Undirected Graphs

A factoring algorithm for computing K-terminal reliability would be equivalent to state space enumeration were we not to make simple probability

reductions as the algorithm proceeds. We recursively apply Equation (2.3), making probability reductions as we go. The edge selection strategy for computing (2.3) attempts to avoid creating subgraphs with irrelevant edges.

Figure 5 shows the binary computational tree resulting from using the factoring algorithm for the two-terminal problem corresponding to the graph at the top of Figure 5 with distinguished nodes s and t, so that $K = \{s,t\}$. In the initial step, we pivot on edge e forming two subgraphs: G_e, corresponding to e working, and $G - e$, corresponding to e failed. Parallel and series probability reductions are now possible. In a parallel probability reduction, two edges, say 2 and 3 in G_e, are replaced by a single edge with associated probability $p_2 + p_3 - p_2p_3$. Likewise in G_e, this new edge with edge 5 are in series. The two edges in series are then replaced by a single edge with associated reliability $(p_2 + p_3 - p_2p_3)p_5$. (In Figure 5, the parallel and series reductions are not shown.) Pivoting now proceeds on edge 4 resulting in two additional subgraphs, each of which can be reduced to a single edge by series and parallel probability reductions. The "leaves" of the tree are the four subgraphs at the bottom of the tree.

If each edge i has probability p of working, it can be shown using the binary computational tree (illustrated in Figure 5) that the system reliability in this case is

$$\begin{aligned}
R_K(G) &= p^2(((((p \amalg p)p) \amalg p)p) \amalg p) + p(1-p)(((p \amalg p)p)(p^2 \amalg p)) \\
&\quad + p(1-p)((p(p \amalg p)) \amalg (p^2))p + (1-p)^2((p^3 \amalg p)p^2) \\
&= (p^3 + p^4 + p^5 - 5p^6 + 4p^7 - p^8) + (2p^4 - p^5 - 4p^6 + 4p^7 - p^8) \\
&\quad + (3p^4 - 4p^5 - p^6 + 3p^7 - p^8) + (p^3 - 2p^4 + 2p^5 - 3p^6 + 3p^7 - p^8) \\
&= \underline{2p^3 + 4p^4 - 2p^5 - 13p^6 + 14p^7 - 4p^8} .
\end{aligned}$$

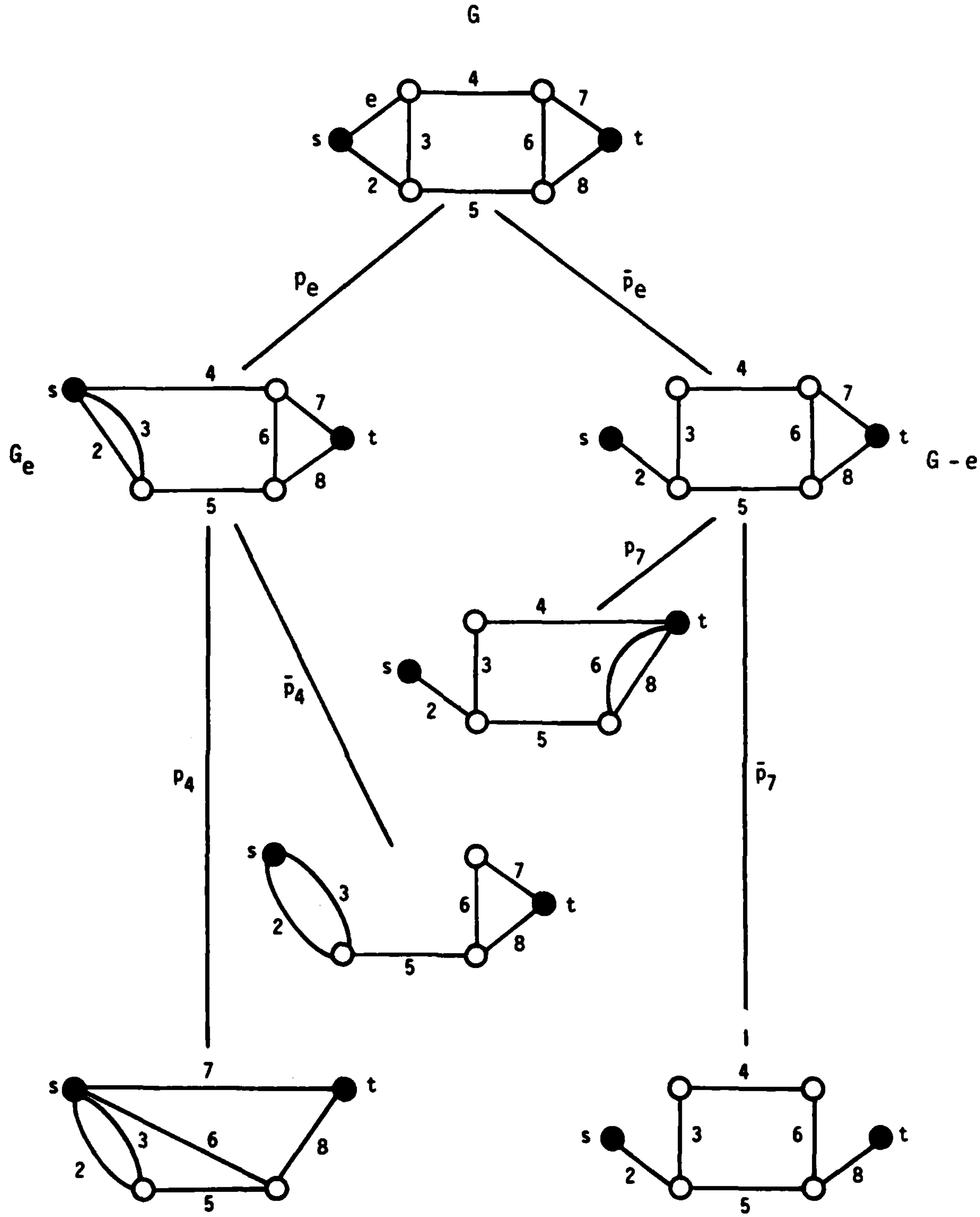


FIGURE 5. BINARY COMPUTATIONAL TREE USING THE FACTORING ALGORITHM

The lower case "ip" operator, $\amalg$, corresponds to calculating the reliability of parallel edges; i.e.,

$$p_i \amalg p_j = p_i + p_j - p_i p_j \; .$$

In Figure 5, $\bar{p}_i = 1 - p_i$. The four graphs at the bottom of the tree are the leaves of the tree and each has domination one since each is series-parallel reducible (Chang, 1981). The domination of the top graph, it turns out, is $D_K(G) = 4$ (the number of "leaves" at the bottom of the tree) and the tree has $2D_K(G) - 1 = 7$ nodes so that the computational running time is proportional to the domination. Satyanarayana and Chang (1983) found that in general the number of leaves in the binary computational tree using a factoring algorithm with series and parallel probability reductions is at least equal to the domination. Using a simple edge selection strategy, they further showed that it is possible to create a backtrack structure which has exactly $D_K(G)$ leaves where $D_K(G)$ is the domination of G . Therefore, this edge selection strategy is optimal for factoring algorithms using series and parallel probability reductions.

Domination Theory for Coherent Systems

A network graph G with distinguished nodes $K \subseteq V$ has a topology defined by the minimal path sets $\mathcal{P} = [P_1, P_2, \ldots, P_p]$. The network may have *both* directed and undirected edges. By definition, all nodes K can communicate if and only if all edges in at least one minimal path set operate. In Barlow and Proschan (1981), a set of edges or components E , and a family of minimal path sets, $\mathcal{P}$, is called a *coherent system*, $(E, \mathcal{P})$, if $\mathcal{P} = [P_1, P_2, \ldots, P_p]$ is a minimal family, i.e., no P_i is contained in another member of the family *and* $E = \bigcup_{i=1}^{p} P_i$. Coherent systems include

all network graphs as well as logic trees (or fault trees) *without not* gates. A k-out-of-n system with $n > 2$ and $1 < k < n$ is a coherent system which *cannot* be represented as a network (unless replicated edges are allowed).

The two-terminal network in Figure 2 will be used to illustrate ideas. For this example, $E = \{1,2,3,4,e\}$ while $\mathcal{P} = [\{1,3\},\{1,2,4\},\{e,4\},\{e,2,3\}]$.

By pivoting on component $e \in E$, we create two subsystems, corresponding to the system with e failed and to the system with e perfect, respectively. Let $\mathcal{P}(e) = [P_i \mid e \in P_i$ and $P_i \in \mathcal{P}]$ and $\mathcal{P}(e') = [P_i \mid e \notin P_i$ and $P_i \in \mathcal{P}]$. Then

$$\mathcal{P} = \mathcal{P}(e) \cup \mathcal{P}(e') .$$

In our example, $\mathcal{P}(e) = [\{e,4\},\{e,2,3\}]$ and $\mathcal{P}(e') = [\{1,3\},\{1,2,4\}]$. In all cases, $\bigcup_{P_i \in \mathcal{P}(e')} P_i \subseteq E - e$. In our example, $\bigcup_{P_i \in \mathcal{P}(e')} P_i = E - e$ so that $(E - e,\mathcal{P}(e'))$ is coherent and corresponds to our system with e failed. If $\bigcup_{P_i \in \mathcal{P}(e')} P_i \subset E - e$, then $(E - e,\mathcal{P}(e'))$ would have *no* formations so that in this case $d(E - e,\mathcal{P}(e')) = 0$ and $(E - e,\mathcal{P}(e'))$ would be noncoherent. (In this setup, $d(E,\mathcal{P})$ denotes signed domination.)

To describe a system with e perfect, let

$$\mathcal{P} - e = [P_1 - e,P_2 - e, \ldots, P_p - e] .$$

If $e \notin P_i$, then P_i is included as it is. Let $M[\mathcal{P} - e]$ be the set minimization of $\mathcal{P} - e$. In our example,

$$\mathcal{P} - e = [\{1,3\},\{1,2,4\},\{4\},\{2,3\}]$$

and

$$M[\mathcal{P} - e] = [\{1,3\},\{4\},\{2,3\}]$$

since $\{4\} \subset \{1,2,4\}$. In this case,

$$\bigcup_{A_i \in \mathcal{M}[\mathcal{P}-e]} A_i = E - e ,$$

so that $(E - e, \mathcal{M}[\mathcal{P} - e])$ corresponds to our example system with e perfect. In general, we only know that

$$\bigcup_{A_i \in \mathcal{M}[\mathcal{P}-e]} A_i \subseteq E - e$$

so that $(E - e, \mathcal{M}[\mathcal{P} - e])$ might be noncoherent.

The following signed domination theorem is proved in Barlow (1982). It was first proved by Satyanarayana and Chang (1983) for undirected networks.

Theorem 2.0: (Signed Domination Theorem)

For any coherent system $(E,\mathcal{P})$ and $e \in E$,

$$d(E,\mathcal{P}) = d(E - e, \mathcal{M}[\mathcal{P} - e]) - d(E - e, \mathcal{P}(e')) .$$

In our example, $d(E - e, \mathcal{M}[\mathcal{P} - e]) = 1$ and $d(E - e, \mathcal{P}(e')) = -1$ so that $d(E,\mathcal{P}) = 2$.

Using Theorem 2.0 and induction, it is easy to verify the following corollary for *undirected* networks.

Corollary:

For a coherent system corresponding to a K-terminal undirected network problem,

$$d(E,\mathcal{P}) = (-1)^{n-v+1} D(E,\mathcal{P})$$

where $D(E,\mathcal{P}) = |d(E,\mathcal{P})|$ is the domination, n is the number of edges and v is the number of vertices or nodes.

Satyanarayana and Chang (1983) proved the following domination theorem.

<u>Theorem 2.1</u>: (Domination Theorem)

For any K-terminal *undirected* graph,

$$D(E,\mathcal{P}) = D(E - e, M[\mathcal{P} - e]) + D(E - e, \mathcal{P}(e')) \ .$$

Proof:

By Theorem 2.0,

$$d(E,\mathcal{P}) = d(E - e, M[\mathcal{P} - e]) - d(E - e, \mathcal{P}(e')) \ .$$

By the corollary to Theorem 2.0,

$$d(E,\mathcal{P}) = (-1)^{n-v+1} D(E,\mathcal{P})$$

if the undirected graph has n edges and v nodes. The subgraph corresponding to e working has signed domination

$$d(E - e, M[\mathcal{P} - e]) = (-1)^{(n-1)-(v-1)+1} D(E - e, M[\mathcal{P} - e])$$

since e has been contracted and its two end vertices merged into one. Similarly, the subgraph corresponding to e failed has signed domination

$$d(E - e, \mathcal{P}(e')) = (-1)^{(n-1)-v+1} D(E - e, \mathcal{P}(e'))$$

since only edge e is deleted. Hence, by Theorem 2.0,

$$\begin{aligned} d(E,\mathcal{P}) &= (-1)^{(n-1)-(v-1)+1} D(E - e, M[\mathcal{P} - e]) \\ &\quad - (-1)^{(n-1)-v+1} D(E - e, \mathcal{P}(e')) \\ &= (-1)^{(n-1)-(v-1)+1} [D(E - e, M[\mathcal{P} - e]) + D(E - e, \mathcal{P}(e'))] \ . \end{aligned}$$ Q.E.D.

The domination theorem is in general not true for directed graphs. For example, the domination of the cyclic directed graph in Figure 3 is zero.

(In fact, all directed networks with cycles have domination zero [cf. R. R. Willie (1980)].) However, if we pivot on e, we obtain a subgraph corresponding to e deleted which has positive domination. Hence, the domination theorem is not true for this example.

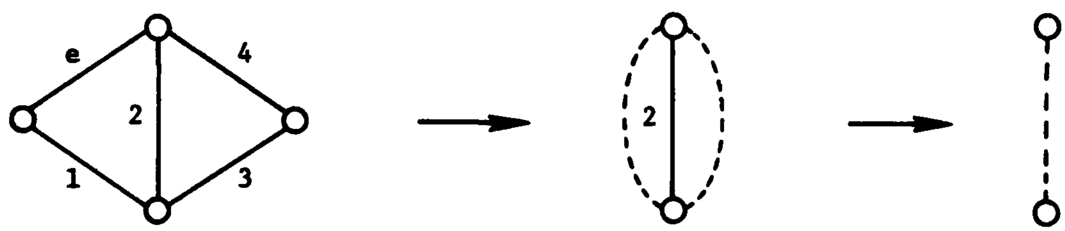
From Theorem 2.1, it can be shown that the computational complexity of any factoring algorithm for undirected networks based on pivoting and parallel and series reductions will require at least $2D(E,\mathcal{P}) - 1$ steps. (Every binary computational tree has number of nodes equal to twice the number of leaves minus one.) By using an edge selection strategy for which both $D(E - e,M[\mathcal{P} - e]) > 0$ and $D(E - e,\mathcal{P}(e')) > 0$, the number of steps is exactly $2D(E,\mathcal{P}) - 1$. This is illustrated by Figure 5.

The *factoring algorithm*, namely pivoting followed by parallel and series probability reductions and repeated until only single edges or "K trees" are obtained, can be applied to any coherent system reliability problem. (A *K-tree* of a graph G with respect to K is any minimal graph which connects all the distinguished nodes in K.) However, the optimal edge selection strategy is only known because of the domination theorem. The domination itself is mainly of theoretical interest. In the worst case, the domination will be exponential in the number of edges and would normally *not* be computed. It does, however, offer a sort of theoretical benchmark. For example, if we delete all arrows from a directed graph, the resulting undirected graph will have a domination value which will provide an upper bound on the best factoring algorithm for the directed network. This is so because the domination of the two subgraphs created by pivoting on an edge in the undirected graph will be at least as much as the domination of the undirected graphs corresponding to the subgraphs obtained by pivoting on the same edge in the directed graph.

3. SPECIAL STRUCTURES

Although the factoring algorithm can, in principle, solve all reliability problems, it is, in the worst case, an exponential time algorithm. For very large networks, we need linear or polynomial time algorithms in order to calculate system reliability in "reasonable" computing time. By introducing additional probability reductions, such algorithms have been found for both directed and undirected network graphs of special structure.

An undirected graph $G = (V,E)$ is said to be *basically series-parallel* if the graph (without distinguished nodes) can be reduced to a single edge by series and parallel *replacements*. A *replacement* as opposed to a probability *reduction* does *not* involve the probability measure which may be associated with the graph. For example, Figure 2 can be reduced to a single edge by series and parallel replacements as follows:

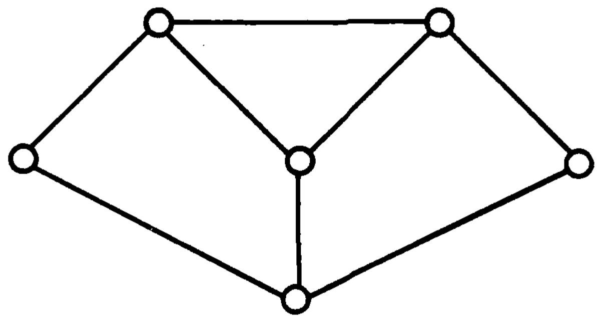


SERIES AND PARALLEL REPLACEMENTS

FIGURE 6

Note that vertices s and t are no longer distinguished. Edges e and 1 are replaced by a dotted line using a series replacement as are edges 3 and 4. Finally, the remaining three edges in parallel are replaced by a single edge. No probability calculations are involved.

The network in Figure 1 on the other hand is *not* basically series-parallel. Neither is the network in Figure 7.



EXAMPLE NETWORK WHICH IS NOT *BASICALLY SERIES-PARALLEL*

FIGURE 7

A *directed* network is *basically series-parallel* if the underlying undirected graph (without arrows on edges) is *basically series-parallel*. For example, the network in Figure 3 is basically series-parallel.

A. Satyanarayana and R. K. Wood (1982) provide linear time algorithms for calculating the K terminal reliability of undirected networks which are basically series-parallel. They introduce probability reductions called polygon-to-chain reductions to accomplish this.

A. Agrawal and A. Satyanarayana (1983) provide linear time algorithms for calculating the source to K terminal reliability of *rooted, directed* networks which are *basically series-parallel*. One node in K is designated the *root* and the reliability problem is to calculate the probability that the root can communicate with the remaining $K \subseteq V$ vertices.

The Minimal Domination of Undirected Graphs

The minimal domination $M(G)$ of a graph G is defined by

$$M(G) = \underset{K:|K|=2}{\text{minimum}}\; D_K(G)$$

where K is a distinguished set of nodes of G. While the domination $D_K(G)$ depends on *both* the graph G and the distinguished set of nodes K, $M(G)$ obviously depends only on G. Whereas $D_K(G) = 1$ if and only if G is reducible to a "K-tree" by series and parallel probability reductions, [Satyanarayana and Chang (1983)], $M(G) = 1$ if and only if G is *basically series-parallel* [R. K. Wood (1982)]. Thus, the graph at the top of Figure 5 has $D_K(G) = 4$ where $K = \{s,t\}$, but $M(G) = 1$ since G is *basically series-parallel.* Using series and parallel probability reductions and the polygon-to-chain reduction in Table 1, $R_K(G)$ can, in this case, be computed *without* pivoting so that a linear time algorithm exists for this problem and in fact for all such problems where G is *basically series-parallel.* There are 6 additional polygon-to-chain reductions necessary to treat cases where $|K| > 2$. [See R. K. Wood (1982).]

Figure 8 provides an example graph where the domination is $D_K(G) = 2^{(|E|-2)/3}$ but $M(G) = 1$.

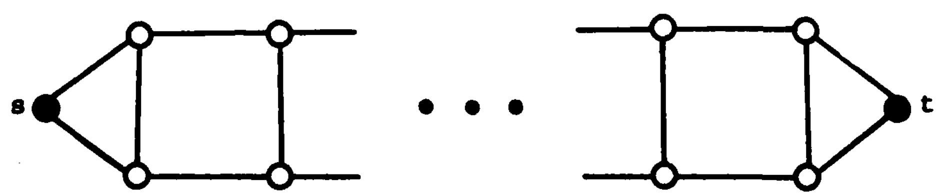
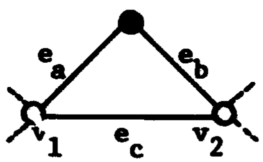
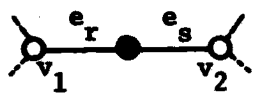


FIGURE 8

TABLE 1

POLYGON-TO-CHAIN REDUCTION*

Polygon Type	Chain Type Replacement	Reduction Formulas	New Edge Reliabilities
		$\alpha = q_a p_b q_c$	$p_r = \dfrac{\delta}{\alpha+\delta}$
		$\beta = p_a p_b p_c$	$p_s = \dfrac{\delta}{\beta+\delta}$
		$\delta = p_a p_b p_c\left(1 + \dfrac{q_a}{p_a} + \dfrac{q_b}{p_b} + \dfrac{q_c}{p_c}\right)$	$\Omega = \dfrac{(\alpha+\delta)(\beta+\delta)}{\delta}$

*Note: Darkened vertices represent K-vertices.

The system reliability, $R_K(G)$, is equal to the reliability of the new system with the chain replacement times Ω .

The minimum domination theorem was proved by R. Procesi-Ciampi (1981). [See also Satyanarayana and Procesi-Ciampi (1981).]

Theorem 3.1: (Minimum Domination Theorem)

For any *undirected* graph $G = (V,E)$,

$$M(G) = M(G_e) + M(G - e) .$$

R. K. Wood (1982) used this and other properties of minimum domination to evaluate the computational complexity of undirected networks relative to pivoting and polygon-to-chain reductions.

4. CONCLUSION

A theoretical breakthrough occurred in 1978 with the publication by Satyanarayana and Prabhakar which first introduced the idea of domination into network reliability. This paper was concerned with directed graphs and suggested that directed cyclic graphs have domination zero, a result rigorously proved by R. R. Willie (1980). In 1981, Satyanarayana and Chang (see their 1983 paper) first noticed the connection between domination theory and the computational complexity of factoring algorithms. This was followed by the notion of minimal domination and the development of linear time algorithms for special structure graphs [cf. R. K. Wood (1982)].

The probability measure associated with the graphs in the literature surveyed is extremely simplistic and unrealistic for many practical network reliability problems. However, a theoretical basis now exists for more research into realistic probability measures. At the present time, there is no comparable theoretical basis for analyzing the computational complexity of logic (or fault) trees. These structures are perhaps more useful in practice than networks. Further research is badly needed in this field.

REFERENCES

Abraham, J. A. (1979), "An Improved Algorithm for Network Reliability," IEEE Transactions on Reliability, Vol. R-28, pp. 58-61.

Aggarwal, K. K., K. B. Misra and J. S. Gupta (1975), "A Fast Algorithm for Reliability Evaluation," IEEE Transactions on Reliability, Vol. R-24, pp. 83-85.

Agrawal, A. (1982), "Reliability Analysis of Rooted Directed Networks," Ph.D. Thesis, Operations Research Center, University of California, Berkeley.

Agrawal, A. and A. Satyanarayana (1982), "An $O(|E|)$ Time Algorithm for Computing the Reliability of a Class of Directed Networks," ORC 82-7, Operations Research Center, University of California, Berkeley.

Agrawal, A. and A. Satyanarayana (1983), "Source-to-K-Terminal Reliability Analysis of Rooted Communication Networks," ORC 83-2, Operations Research Center, University of California, Berkeley.

Ball, M. O. (1977), "Network Reliability Analysis: Algorithms and Complexity," Ph.D. Thesis, Department of Operations Research, Cornell University.

Ball, M. O. (1980), "Complexity of Network Reliability Computations," Networks, Vol. 10, pp. 153-165.

Barlow, R. E. and F. Proschan (1981), STATISTICAL THEORY OF RELIABILITY AND LIFE TESTING, To Begin With, Silver Spring, MD.

Barlow, R. E. (1982), "Set Theoretic Signed Domination for Coherent Systems," ORC 82-1, Operations Research Center, University of California, Berkeley.

Buzzcott, J. A. (1980), "A Recursive Algorithm for Finding Reliability Measures Related to the Connection of the Nodes of a Graph," Networks, Vol. 10, pp. 311-327.

Chang, M. K. (1981), "A Graph Theoretic Appraisal of the Complexity of Network Reliability Algorithms," Ph.D. Thesis, Operations Research Center, University of California, Berkeley.

Deo, N. (1974), GRAPH THEORY WITH APPLICATIONS TO ENGINEERING AND COMPUTER SCIENCE, Prentice-Hall, New Jersey.

Fratta, L. and U. G. Montanari (1973), "A Boolean Algebra Method for Computing the Terminal Reliability in a Communication Network," IEEE Transactions on Circuit Theory, Vol. C-20, pp. 203-211.

Gabow, H. N. and E. W. Myers (1978), "Finding all Spanning Trees of Directed and Undirected Graphs," SIAM Journal on Computing, Vol. 7, pp. 280-287.

Garey, M. R. and D. S. Johnson (1979), COMPUTERS AND INTRACTABILITY, W. H. Freeman and Company, San Francisco, CA.

Hagstrom, J. N. (1980), "Combinatoric Tools for Computing Network Reliability," Ph.D. Thesis, Operations Research Center, University of California, Berkeley.

Jerrum, M. (1981), "On the Complexity of Evaluating Multivariate Polynomials," Ph.D. Thesis and Technical Report CST-11-81, Department of Computer Science, University of Edinburgh.

Lee, S. H. (1980), "Reliability Evaluation of a Flow Network," IEEE Transactions on Reliability, Vol. R-29, pp. 24-26.

Misra, K. B. (1970), "An Algorithm for the Reliability Evaluation of Redundant Networks," IEEE Transactions on Reliability, Vol. R-19, pp. 146-151.

Moskowitz, F. (1958), "The Analysis of Redundancy Networks," AIEE Transactions on Communications and Electronics, Vol. 77, Part I, pp. 627-632.

Murchland, J. D. (1975), "Fundamental Concepts and Relations for Reliability Analysis of Multi-state Systems," in RELIABILITY AND FAULT TREE ANALYSIS, R. E. Barlow, J. B. Fussell and N. D. Singpurwalla (editors), SIAM, PA, pp. 581-618.

Procesi-Ciampi, R. (1981), "A Minimality Property for Acyclic Orientations," unpublished Operations Research Center Report, University of California, Berkeley.

Provan, J. S. and M. O. Ball (1981), "The Complexity of Counting Cuts and of Computing the Probability That a Graph Is Connected," Working Paper MS/S 81-002, University of Maryland, MD.

Rosenthal, A. (1974), "Computing Reliability of Complex Systems," Ph.D. Thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley.

Rosenthal, A. (1975), "A Computer Scientist Looks at Reliability Computations," in RELIABILITY AND FAULT TREE ANALYSIS, R. E. Barlow, J. B. Fussell and N. D. Singpurwalla (editors), SIAM, PA, pp. 133-152.

Rosenthal, A. and D. Frisque (1977), "Transformations for Simplifying Network Reliability Computations," Networks, Vol. 7, pp. 97-111.

Satyanarayana, A. and A. Prabhakar (1978), "New Topological Formula and Rapid Algorithm for Reliability Analysis of Complex Networks," IEEE Transactions on Reliability, Vol. R-27, pp. 82-100.

Satyanarayana, A. and R. Procesi-Ciampi (1981), "On Some Acyclic Orientations of a Graph," ORC 81-11, Operations Research Center, University of California, Berkeley.

Satyanarayana, A. (1982), "A Unified Formula for the Analysis of Some Network Reliability Problems," IEEE Transactions on Reliability, Vol. R-31, pp. 23-32.

Satyanarayana, A. and R. K. Wood (1982), "Polygon-to-Chain Reductions and Network Reliability," ORC 82-4, Operations Research Center, University of California, Berkeley.

Satyanarayana, A. and M. K. Chang (1983), "Network Reliability and the Factoring Theorem," Networks, Vol. 13, pp. 107-120.

Willie, R. R. (1978), "Computer-Aided Fault Tree Analysis," ORC 78-14, Operations Research Center, University of California, Berkeley.

Willie, R. R. (1980), "A Theorem Concerning Cyclic Directed Graphs with Applications to Network Reliability," Networks, Vol. 10, pp. 71-78.

Wood, R. K. (1982), "Polygon-to-Chain Reductions and Extensions for Reliability Evaluations of Undirected Networks," Ph.D. Thesis, Operations Research Center, University of California, Berkeley.

END

FILMED

10-83

DTIC